UDC: 004.932 https://doi.org/10.15407/jai2025.03.103

D. Sanzharov[1], **T. Filimonova**[2]
[1, 2]State University of Trade and Economics, Ukraine
19, Kyoto st., Kyiv, 02156
[1]d.sanzharov_fit_4m_25_m_d@knute.edu.ua
[2]t.filimonova@knute.edu.ua
[1]https://orcid.org/0009-0004-6600-5557
[2]https://orcid.org/0000-0001-9467-0141

LICENCE PLATE RECOGNITION AND VEHICLE IDENTIFICATION US-ING DEEP LEARNING

Abstract. This paper presents a methodology for automated licence plate recognition and vehicle identification designed for Ukrainian standards. The approach combines YOLOv11 object detection with EasyOCR text recognition, incorporating deskewing and adaptive thresholding preprocessing techniques. A custom dataset of 467 images containing Ukrainian DSTU-compliant licence plates was created from the Auto. RIA platform. The system employs format-specific error correction and character substitution rules. Experimental results demonstrate an F_1 score of 0.92 at IoU@0.5 for detection tasks, with performance variations across vehicle classes revealing challenges in commercial vehicle recognition and motorcycle licence plate processing. This work contributes annotated dataset for Ukrainian licence plate recognition and provides a practical foundation for automated traffic monitoring systems.

Keywords. Deep learning, YOLO, EasyOCR, deskewing, computer vision, recognition, identification.

Introduction

Modern developed countries are defined by high automobile ownership rates, both personal and commercial. As countries develop, the ownership increases. Road vehicles require a lot of infrastructure, and many individually controlled vehicles present an issue for management and safety.

To manage road network efficiently and safely, authorities require large quantities of data to make quality decisions or manage the system in real time. One of the crucial tools to collect that data is surveillance using cameras directed at the road. However, the amount of data being collected would be impractical to process manually thus requiring automation.

The *relevance* of this paper is in presenting a methodology to help with this automation problem, but not with completely solving it. The methodology involves licence place recognition and identification of road vehicles using YOLO (You Only Look Once), EasyOCR, deskewing program and OpenCV for Ukrainian formats specifically.

The *object* of this paper is processes of recognition and identification using deep learning and general image processing techniques.

The *subject* of this paper is models, methods, and informational technologies in systems of licence plate recognition and road vehicle identification.

The *approbation results* were presented in student scientific and practical conference "Information technologies and cybersecurity in wartime" in 2025 [1].

Literature review

The topic of automated licence plate recognition (ALPR) has been explored many times using different methods and architectures. This popularity is caused by the relevance of this issue. However, all explored papers and solutions are focusing on specific countries or regions. This means that existing methods are rarely fully applicable without alterations. Considering the amount of available papers, it was chosen to review only three works here.

The *first* paper [2] presents a methodology based on YOLOv8 and EasyOCR for licence plate recognition with high accuracy. However, their recognition is limited to only recognizing countries of origin with European style licence plates, i.e. narrow rectangles with standardized county identifier positions.

The methodology consists of steps:

1. Licence plate detection using YOLOv8 model trained on custom image dataset.
2. Crop the image to the area of detected licence plate.
3. Image preprocessing:
   i. Color clustering using K-means.

ii. Image binarization using clusters.

iii. Morphological operations to improve text clarity.

4. Use EasyOCR to recognize text identifiers for countries.

The authors claim F_1 of 99% for licence plate detection, but they do not mention IoU threshold, thus the detection is "positive" as long as the plate is fully in the bounding box. The text recognition character-level-accuracy score is 99.95%.

The *second* paper [3] presents a methodology based on R-CNN for recognition of People's Republic of China licence plates. It is designed to work in various environments and was trained on a custom dataset of non-uniform images that vary in resolution and environment.

The methodology consists of steps:

1. Licence plate detection using R-CNN.
2. Text recognition:

i. Plate rectification based on estimated corners from previous step.

ii. Character segmentation.

iii. Each character passed to a recognizer.

Image rectification is possible here because R-CNN used estimates corners' position, unlike YOLO framework that provides only bounding box. The authors decided to use a custom text recognition using their own segmentation and classification because general purpose OCR solutions at the time of publishing had unreliable results in difficult environments. Claimed accuracy at IoU@0.5-0.9 is 0.7813, however at IoU@0.5 accuracy is 0.9668, which is still practical for this use. Accuracy of text recognition is at 0.8905 and there is a clear difference in accuracy with and without licence plate rectification. Accuracy score improves to 0.9510 when at least 1 error is tolerated, meaning the method can be improved with better error correction or recognizer.

The *third* paper [4] presents a methodology based on YOLOv2 and Fast-YOLO detection and CNN-based text recognition for Brazilian licence plates. Their model is trained on their own UFPR-ALPR dataset that consists of images obtained from video recorded inside driving car.

The methodology consists of steps:

1. Vehicle detection and cropping the image to a bounding box.
2. Licence plate detection and cropping to a bounding box.
3. Character segmentation and per-character recognition.
4. Multi-frame recognition.
5. Error correction using majority vote from temporal data.

For the text recognition CNN was used like in the *second* paper and trained to work specifically with the character set and forms used by Brazil. The idea behind error correction is using temporal information by recognizing the same licence plate over multiple frames of video. The general assumption is most frequent character at certain position is the true character and others are false positives of the OCR.

For the vehicle detection YOLOv2 was used and its F_1 score is 0.99, however it was achieved at very low confidence threshold of 0.125.

For the licence plate detection Fast-YOLO is used, because the difference is <0.5% and the authors prioritized lower computational cost. Recall rate is at 0.98 and no other metric is provided, however only one road vehicle's licence plate was not detected through 30 frames of video.

The proposed character recognition has achieved 64.89% recognition rate when processing frames individually and 78.33% when using temporal redundancy. However, the authors stated that there is a consistent issue with recognizing motorcycles.

Methodology

We propose a methodology that is inspired by the [2–4]. Detection of both licence plates and road vehicles is handled by YOLOv11 model [5] in a single shot. It is the most recent version available at the time of writing and unlike earlier versions discussed in previous papers, it combines CNN and Visual Transformer architectures for a better accuracy. Paper [6] provides a good review of evolution of the YOLO framework, despite being focused on application in the textile industry.

For the OCR EasyOCR [7] was chosen due its accuracy shown in [2] and in [8] it was found to work better in noisy and distorted environment. Before OCR is executed, image is preprocessing using methods from OpenCV [9], an open-source computer vision library.

```
def enhance_licence_plate(warped_plate):
  gray = cv2.cvtColor(warped_plate, cv2.COLOR_BGR2GRAY)
  adaptive = cv2.adaptiveThreshold(
    gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 15, 5)
  _, otsu_adaptive = cv2.threshold( adaptive, 0, 255, cv2.THRESH_BINARY +
    cv2.THRESH_OTSU)
 return otsu_adaptive
```

Listing 1. OpenCV thresholding function

Unlike methodology in [3], we lack estimation of corners of the licence plate and rectification had a significant influence on the quality. To replace rectification, open source technology [10] was selected to deskew the image and partially fix the perspective. After being deskewed, OpenCV Adaptive Threshold (the function in Listing 1) is used to improve OCR accuracy. The example of preprocessing can be seen in Figure 1.

Figure 1. Preprocessing demonstration

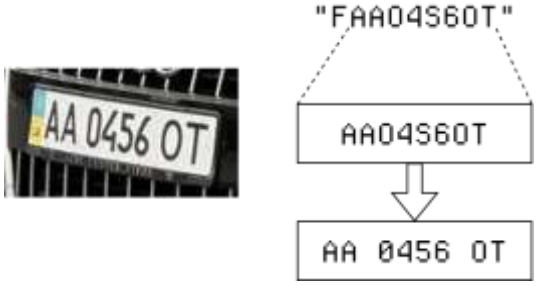

Figure 2. Error correction demonstration

The methodology works with a still images, thus we cannot use temporal redundancy like in [4]. So, for error correction a simple algorithm was chosen. EasyOCR outputs a string of text that contains recognized line of text, however it often confuses characters like 0 and O, or 1 and I. Also, sometimes it gives false positives for characters that aren't there. The fix for this is pattern matching licence plate format as defined by DSTU [11] and implemented using Python RegEx standard library [12]. Then alphabetic and numeric characters are switched to the closely resembling alternative as it can be seen in Figure 2.

Dataset collection

For the model training we decided to create a custom dataset with road vehicles from Ukraine. As a source for the images it was decided to use popular used cars sale platform Auto.RIA [13]. It was chosen because it hosts a large quantity of different road vehicles of different models and years. Also, it is common to have a wide selection of photos from different angles, which helps with model training. Example images can be seen in Figure 3.

Figure 3. Examples from the dataset

Every image was selected and annotated manually. The following classes were defined for the object classification task: car, truck, motorbike, licence plate. In total there are 467 images, 420 licence plates, 287 cars, 175 trucks, 57 motorbikes. It should be noted that truck class also includes vans, and motorbike class includes both two- and three- wheeled options.

```
import subprocess
def deskew_text_validated(image_path):
  command = [DESKEW, '-o', 'deskewed.png', image_path]
  try:
    _ = subprocess.run(command, capture_output=True, text=True, check=True)

    deskewed = cv2.imread('deskewed.png')
    return deskewed
  except subprocess.CalledProcessError as e:
    print(f"Error running deskew: {e}")
  return cv2.imread(image_path)
```

Listing 3. Deskewing function

```
OCR_READER = easyocr.Reader(['en'])
def extract_text(img):
  res = OCR_READER.readtext(img, allowlist='ABCDEHIKMNOPTXYZJ0123456789|', text_threshold=0.3)
  return [''.join(p) for p in permutations([r[1] for r in res])]

def correct_chars(text):
  RULES = {
    'outer': {'0':'O','1':'I','8':'B','2':'Z','7':'Z','|':'I'},
    'inner': {'O':'0','I':'1','B':'8','Z':'7','|':'1'}
  }
  c = list(text)
  for i in range(len(c)):
    key = 'outer' if i in [0,1,6,7] else ('inner' if i in [2,3,4,5] else None)
    if key: c[i] = RULES[key].get(c[i], c[i])
  return ''.join(c)
```

Listing 4. EasyOCR recognition and error correction function

Experimental implementation

Python 3.12.6 was used for development purposes. It was decided to use YOLOv11 in Nano configuration from Ultralitics [5] due its lower computational cost. The configuration of model training in Listing 2 is simplistic. The only notable configurations are early stop at 20 epochs, cosine decay of learning rate, and dataset augmentation.

From the point of view of the experimental implementation, deskewing method is a black box and is executed as a separate process as seen in Listing 3.

EasyOCR sometimes recognizes text and separate strings. We check all permutations of these strings to find possible licence plate. EasyOCR and error correction code can be seen in Listing 4.

Results

The YOLO model responsible for the identification of road vehicles and licence plates has a high F_1 score at IoU@0.5 of 0.92 at confidence threshold of 0.682 as it can be seen in Figure 4.

```
from ultralytics import YOLO
model = YOLO("yolo11n.yaml")

results = model.train(
data='data.yaml', imgsz=640,
patience=20, cos_lr=True,
augment=True, device=0,
project='vehicle_detection',
name='train_nano'
)
```

Listing 2. YOLOv11 Nano training

We also can see that almost at every confidence threshold F_1 for licence plates is higher than the overall level. Also, motorbikes have a high recognition rate, but at higher confidence threshold it rapidly drops to 0. For the most part, trucks have the worst recognition rate, this might be caused by grouping several different road vehicle types with different features under a single class, bringing the average down.

The recognition rate for licence plates includes road vehicles that were not identified because the pipeline will process detected licence plates anyway.

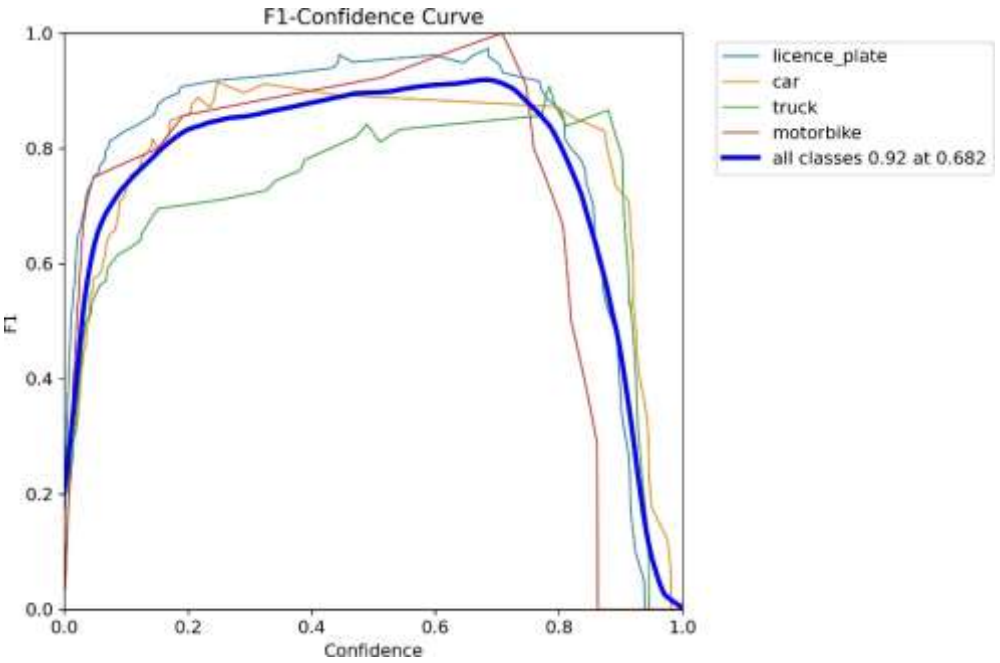

Figure 4. F_1-Confidence plot

The recognition rate overall and by road vehicle class can be seen in Figure 5. “Correct” is when the recognized plate corresponds to ground truth. “Wrong” is when the recognized plate follows the format, but doesn’t correspond to the ground truth. “Missed” is when plate content wasn’t recognized.

We can see that like in case of identification, trucks have the lowest rates. This is likely caused by inconvenient position of licence plates on most truck models and complex environment. Meanwhile, motorbikes have a significantly higher “wrong” segment. Motorbikes use a more complex licence plates where they have a multi-line text to fit the required information in a narrower space. Since EasyOCR recognizes such text in multiple strings of text, we often receive a text that has a valid format but in the wrong order.

In Figure 6 we can see how many licence plates were wrong or missed because of some errors.

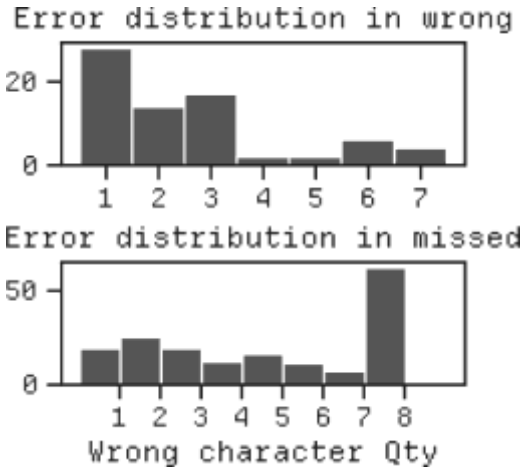

Figure 6. Error distribution histogram

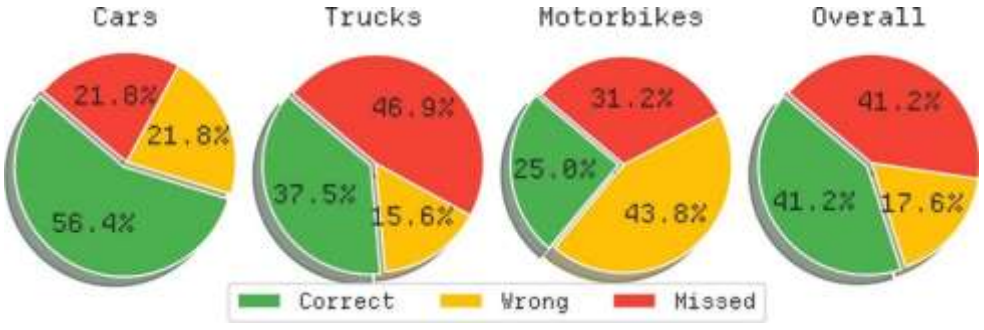

Figure 5. Licence plates recognition rates overall and by class

Discussion

The experimental results demonstrate that YOLOv11 achieves high performance for multi-class vehicle and licence plate recognition with F_1 socre of 0.92 at IoU@0.5, comparable to the results in [2,4] but with more classes. The superior performance of licence plates detection relative to vehicle classification suggests that geometric and content consistency of licence plates makes them more distinct and easier to detect.

The significant performance degradation for truck classification can be attributed to the inconsistent nature of this class, which encompasses vehicles with substantially different visual characteristics (cargo trucks, delivery vans, utility vehicles). This finding suggests that future work should implement hierarchical classification or separate specialized models for commercial vehicle subtypes.

The higher error rate for motorcycle licence plates stems from their multi-line format, which creates ambiguity in character sequence reconstruction when EasyOCR segments text into multiple strings. This represents a fundamental limitation of treating OCR output as independent character recognitions rather than structured sequence prediction. The permutation-based approach partially addresses this but introduces computational overhead and potential false positive matches.

The error correction methodology using DSTU format validation and character substitution rules demonstrates practical effectiveness but remains heuristic rather than principled. The observed confusion between visually similar characters (O/0, I/1, B/8) reflects inherent limitations in single-frame recognition without contextual information. Unlike [4], which leverages temporal redundancy across video frames, this approach must rely solely on spatial information, limiting error correction capabilities.

The custom dataset of 467 images represents a unique contribution for Ukrainian ALPR systems, addressing the absence of publicly available datasets for Ukrainian licence plate formats and vehicle types. While smaller than large-scale generic datasets, this domain-specific collection enables training on authentic Ukrainian DSTU-compliant licence plates and local vehicle characteristics, which existing international datasets cannot provide. The observed performance variations across vehicle classes reflect the inherent challenges in domain-specific recognition tasks rather than dataset inadequacy, particularly given the successful F_1 of 0.92 achievement with this specialized training set.

The integration of deskewing as a preprocessing step partially compensates for lack of full rectification method. However, the black box nature of current experimental solution limits its opportunities for optimization.

Conclusions

This work presents a complete methodology for Ukrainian licence plate recognition and vehicle identification using YOLOv11, EasyOCR, and specialized preprocessing techniques. The methodology achieves practical performance levels with F_1 of 0.92 for detection tasks.

Key contributions include:

1. Creation of the first annotated dataset for Ukrainian licence plate recognition with DSTU-compliant formats.
2. Adaptation of state-of-the-art object detection to Ukrainian vehicle characteristics and licensing standards.
3. Development of format-specific error correction using DSTU compliance validation.
4. Demonstration of effective preprocessing strategies for perspective distortion correction in absence of geometric rectification.

The results reveal class-specific performance variations that highlight fundamental challenges in multi-class vehicle recognition, particularly for heterogeneous categories like commercial vehicles. The motorcycle licence plate recognition problem exposes limitations in sequential text reconstruction from segmented OCR output, suggesting future research directions in structured sequence prediction.

The methodology provides a foundation for automated traffic monitoring systems in Ukrainian infrastructure while identifying specific technical challenges that require domain-specific solutions beyond generic computer vision approaches.

References

1. Sanzharov, D. R., & Filimonova, T. O. (2025). Identifikatsiia transportnykh zaso- biv ta nomernykh znakiv z vykorystanniam modeli simeistva Ultralitics YOLO. *Infor- matsiini Tekhnolohii Ta Kiber-bezpeka V Umovakh Voiennoho Chasu*, 6–9.
2. Moussaoui, H., Akkad, N. E., Benslimane, M., El-Shafai, W., Baihan, A., Hewage, C., & Rathore, R. S. (2024). Enhancing automated vehicle identification by inte- grating YOLO v8 and OCR techniques for high-precision license plate detection and recognition. *Scientific Reports*, *14*(1), 14389. https://www.nature.com/articles/s41598-024-65272-1.pdf
3. Dong, M., He, D., Luo, C., Liu, D. & Zeng, W. (2017,). A CNN-Based Approach for Automatic License Plate Recognition in the Wild. *BMVC*. https://bmva-archive.org.uk/bmvc/2017/papers/paper175/paper175. pdf
4. Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. *2018 International Joint Conference on Neural Networks (Ijcnn)*, 1–10. https:// arxiv.org/pdf/1802.09567
5. Ultralitics. *Ultralitics YOLO11 - Ultralitics YOLO Docs*. https://docs.ultralytics.com/ models/yolo11/
6. Mao, M., & Hong, M. (2025). YOLO object detection for real-time fabric defect inspection in the textile industry: A review of YOLOv1 to YOLOv11. *Sensors (Basel, Switzerland)*, *25*(7), 2270. https://doi.org/ 10.3390/s25072270
7. Jaided AI. (2024). *EasyOCR: Ready-to-use OCR with 80+ supported languages and all common writing scripts*. https:// github.com/JaidedAI/EasyOCR
8. Smelyakov, K., Chupryna, A., Darahan, D., & Midina, S. (2021). Effectiveness of Modern Text Recognition Solutions and Tools for Common Data Sources. *COLINS*, 154–165. https://ceur-ws.org/Vol-2870/paper15.pdf
9. OpenCV team. (2024,). *Open Source Computer Vision Library*. https://github. com/opencv/opencv
10. Mauder, M. (2024,). *deskew: Skew cor- rection of images, especially documents*. https://github.com/galfar/deskew
11. Ministry of Economic Development and Trade of Ukraine. *DSTU 4278:2019. Road Transport. Vehicle Registration Plates. General Requirements. Rules of Applica- tion.*
12. Python Software Foundation. (2024). *The Python Language Reference Man- ual* (3.12 ed.). https://docs.python.org/3.12/reference//
13. AUTO.RIA. *Automarket №1, buy and sell cars easily!*. https://auto.ria.com/uk/

The article has been sent to the editors 15.09.25.
After processing 20.09.25.
Submitted for printing 30.09.25.

Copyright under license CCBY-SA4.0.